Technical Cybersecurity

Passwords and Encryption

Passwords

Variety of password encryption techniques

- LANMAN
- NTLM
- UNIX/Linux

Algorithms Too

- MD5
- SHA
- DES
- AES

What are these?

Hashes, one-way, or trapdoor functions

- Bits come in, a unique signature comes out
- Compromised by collision; e.g. two different bit sequences generate the same signature

Examples

- MD5: Compromised, don’t use this
- SHA-256: Still reasonably secure, SHA-1 is not

And the other ones?

Encryption Functions

- AES: Advanced encryption standard
- DES: Data encryption standard

Symmetric v. Asymmetric

- Symmectric algorithms use the same key to encrypt and decrypt
- Asymmetric encryption has two keys; one is treated as public, and the other as private
- Asymmetric methods are slow; usually used to encipher a shared, generated symmetric key for a single session

Encryption is hard!

LANMAN Hash

Windows-centric password storage

- Optional in new versions of windows, but rare
- Stored in Windows *Security Account Manager* database
- Kept for backwards compatibility
- Also stored in various *Active Directory* servers
- Stored as *LANMAN Hashes*

Properties

- Case insensitive
- Not really a hash, uses DES, but called a hash anyway

LANMAN Hash

It works something like this

1. Take a password. We need 14 characters, so if it is less than 14, pad the password.
2. Convert lower case to upper case.
3. Split the password into two 7-character strings.
4. Use the 7-character strings as keys to encrypt the string **KGS!@#$%** with DES.
5. Concatenate the cipher text and store it.

LANMAN Hash

"WORKS" IS RELATIVE I SUPPOSE…

1. Take a password. We need 14 characters, so if it is less than 14, pad the password.
2. Convert lower case to upper case.
3. Split the password into two 7-character strings.
4. Use the 7-character strings as keys to encrypt the string **KGS!@#$%** with DES.
5. Concatenate the cipher text and store it.

NT Hash

MD4 (so this is a real hash).

Other weaknesses?

Neither NT nor LANMAN hashes are salted

- Salting is the process of adding data known by the defender to the data to be hashed or enciphered
- The salt, a randomly generated value, is appended to the data to be hashed, and then the hash is created
- The salt is then stored with the hash
- Makes attacks much more difficult

And on UNIX?